A Generalizable Learning Approach To Accelerate Global Optimization of QCQPs

Harsha Nagarajan

Los Alamos National Laboratory

2026 MIXED INTEGER PROGRAMMING WORKSHOP

May 18, 2026

Funded by Laboratory Directed Research and Development

LA-UR-24-31610

Los Alamos NATIONAL LABORATORY EST.1943

Collaborators

Erin George
(UCSD)

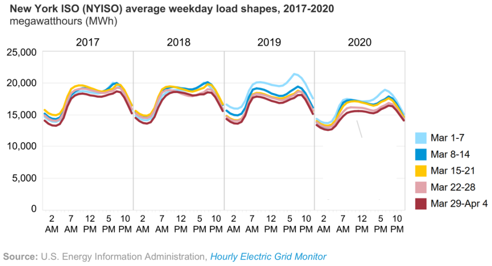
Rohit Kannan
(VTech)

Deepjyoti Deka
(MIT Energy Initiative)

Motivation for Learning

Power grid optimization

Loads

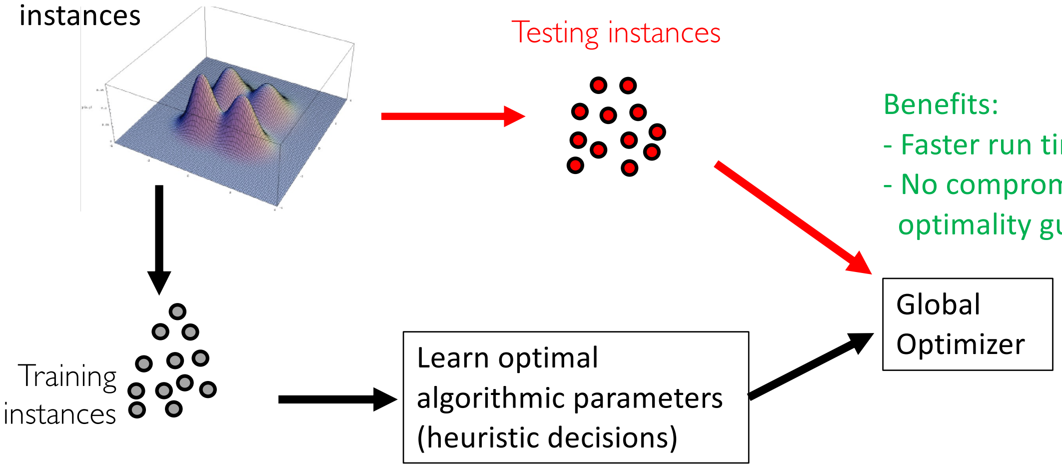


New York ISO (NYISO) average weekday load shapes, 2017-2020

Source: U.S. Energy Information Administration, *Hourly Electric Grid Monitor*

- Power grid operators often solve large set of instances of the same nonconvex problem (AC Optimal Power Flow) with varying problem parameters. The grid topology remains the same while loads, renewable generation, etc. can vary across instances/time periods.
- ML can exploit distribution-specific structure to construct better heuristics decisions via imitation of certain policies.

Motivation for Learning



Distribution of Nonconvex instances

Testing instances

Benefits:
- Faster run times
- No compromise in optimality guarantees

Global Optimizer

Training instances

Learn optimal algorithmic parameters (heuristic decisions)

Literature on learning with optimality guarantees

Branching

[1] Dey, S.S., Han, D. and Wang, Y., 2026. Extreme strong branching for QCQPs. Operations Research Letters, p.107446.

[2] Ghaddar, B., Gómez-Casares, I., González-Díaz, J., González-Rodríguez, B., Pateiro-López, B. and Rodríguez-Ballesteros, S., 2023. Learning for spatial branching: An algorithm selection approach. INFORMS Journal on Computing, 35(5), pp.1024–1043.

[3] González-Rodríguez, B., Alvite-Pazó, R., Alvite-Pazó, S., Ghaddar, B. and González-Díaz, J., 2025. Polynomial optimization: Tightening RLT-based branch-and-bound schemes with conic constraints. JOTA, 204(1), p.12.

[4] González-Rodríguez, B., Gómez-Casares, I., Ghaddar, B., González-Díaz, J. and Pateiro-López, B., 2024. Learning in spatial branching: Limitations of strong branching imitation. arXiv:2406.03626.

Cutting-planes

[5] Baltean-Lugojan, R., Bonami, P., Misener, R. and Tramontani, A., 2019. Scoring positive semidefinite cutting planes for quadratic optimization via trained neural networks. Optimization Online.

[6] González-Rodríguez, B., Alvite-Pazó, R., Alvite-Pazó, S., Ghaddar, B. and González-Díaz, J., 2025. Polynomial optimization: Tightening RLT-based branch-and-bound schemes with conic constraints. JOTA, 204(1), p.12.

[7] Ozen, B. and Kocuk, B., 2025. Learning to relax nonconvex quadratically constrained quadratic programs. arXiv:2501.03954.

Bound tightening / Domain reduction

[8] Cengil, F., Nagarajan, H., Bent, R., Eksioglu, S. and Eksioglu, B., 2022. Learning to accelerate globally optimal solutions to the AC optimal power flow problem. Electric Power Systems Research, 212, p.108275.

[9] Cengil, F., Nagarajan, H., Bent, R., Eksioglu, S. and Eksioglu, B. Learning to accelerate tightening of convex relaxations of the AC optimal power flow problem. Computational Optimization and Applications, pp.761–786, 2025.

[10] Gómez-Casares, I., González-Rodríguez, B., González-Díaz, J. and Rodríguez-Fernández, P., 2024. Impact of domain reduction techniques in polynomial optimization: A computational study. arXiv:2403.02823.

[11] Nannicini, G., Belotti, P., Lee, J., Linderoth, J., Margot, F. and Wächter, A., 2011. A probing algorithm for MINLP with failure prediction by SVM. CPAIOR, Springer, pp.154–169.

[12] Zhang, Y. and Sahinidis, N.V., 2025. Learning to deactivate probing with graph convolutional network for mixed-integer nonlinear programming. Optimization Letters, pp.1–18.

Reformulation

[13] Bonami, P., Lodi, A. and Zarpellon, G., 2018. Learning a classification of mixed-integer quadratic programming problems. CPAIOR, Springer, pp. 595–604.

Open challenge: Develop expert policies that optimize branching variable and break-point selection, with efficient ML models that learn and generalize from them.

Partitioning-based Global Optimization

Making things concrete!

$$\begin{aligned} \min: \quad & f(\boldsymbol{x}) \\ \text{subject to}: \quad & c_i(\boldsymbol{x}) = 0 \quad \forall i \in Eq \\ & h_i(\boldsymbol{x}) \leqslant 0 \quad \forall i \in In \\ & \boldsymbol{x} \in \mathbb{R}^n \end{aligned}$$

Functions *f*, c_i and h_i can be sum of individual quadratic terms of the form:

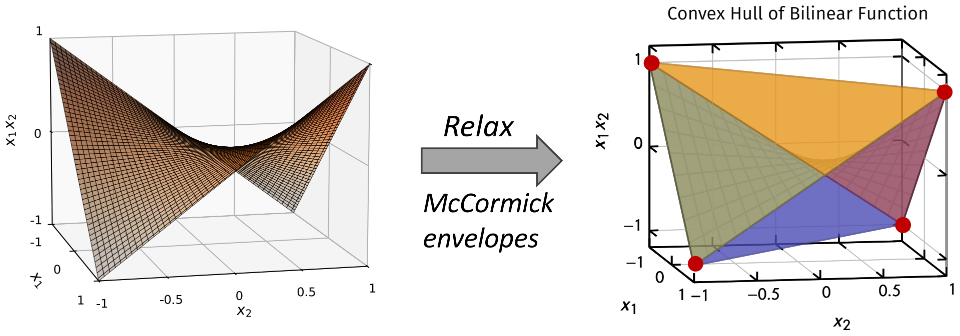
$$\phi(\boldsymbol{x}) = \prod_{i \in \mathcal{I}} x_i. \qquad \phi(\boldsymbol{x}) : [\boldsymbol{l}, \boldsymbol{u}] \longrightarrow \mathbb{R}, \quad |\mathcal{I}| = 2$$

We apply piecewise polyhedral relaxations (PPR) of the graph of $\phi(\boldsymbol{x})$, given by

$$X = \{(\boldsymbol{x}, w) \in [\boldsymbol{\ell}, \boldsymbol{u}] \times \mathbb{R} : w = \phi(\boldsymbol{x})\}$$

Quadratic term relaxations

Quadratic term: $x_1 x_2$



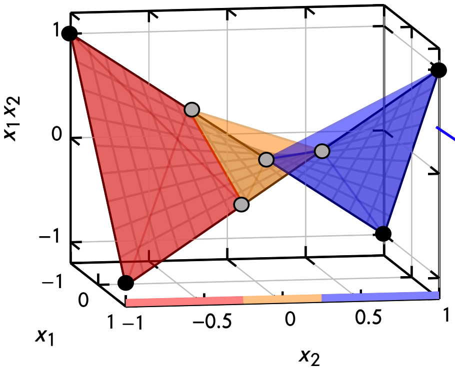
Convex Hull of Bilinear Function

Relax

McCormick envelopes

$$w \geqslant u_2 x_1 + u_1 x_2 - u_1 u_2, \quad w \geqslant \ell_2 x_1 + \ell_1 x_2 - \ell_1 \ell_2,$$
$$w \leqslant u_2 x_1 + \ell_1 x_2 - \ell_1 u_2, \quad w \leqslant \ell_2 x_1 + u_1 x_2 - u_1 \ell_2,$$

Solving QCQPs via Piecewise Polyhedral Relaxations



Variable partitioning

Piecewise McCormick envelopes

$$\textbf{QCQP} \triangleq \min_{x\in\mathbb{R}^n} x^\top Qx + r^\top x$$
$$\text{s.t.:}\quad x^\top Q_i x + r_i^\top x \le b_i \quad i = 1,\dots,m$$
$$x_i \in [\underline{x_i},\ \overline{x_i}] \quad i = 1,\dots,n$$

1. Choose partitioning points and model each quadratic term using Piecewise McCormick envelopes via a MIP formulation to tighten relaxations.
2. Solve a sequence of relaxation MIPs to efficiently approach a near-global optimum for the original QCQP.

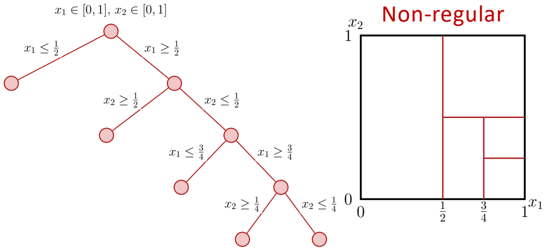
Optimality gap = (UB-LB)/UB

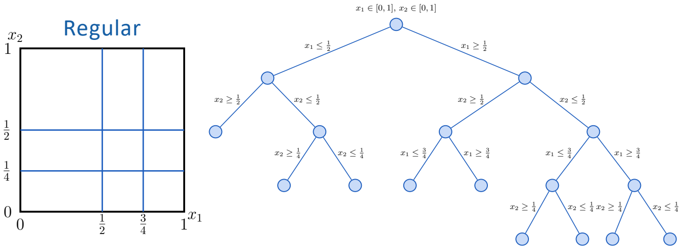
Spatial B&B vs. Partitioning methods

Quadratic term: x_1x_2

Branch-&-Bound search tree

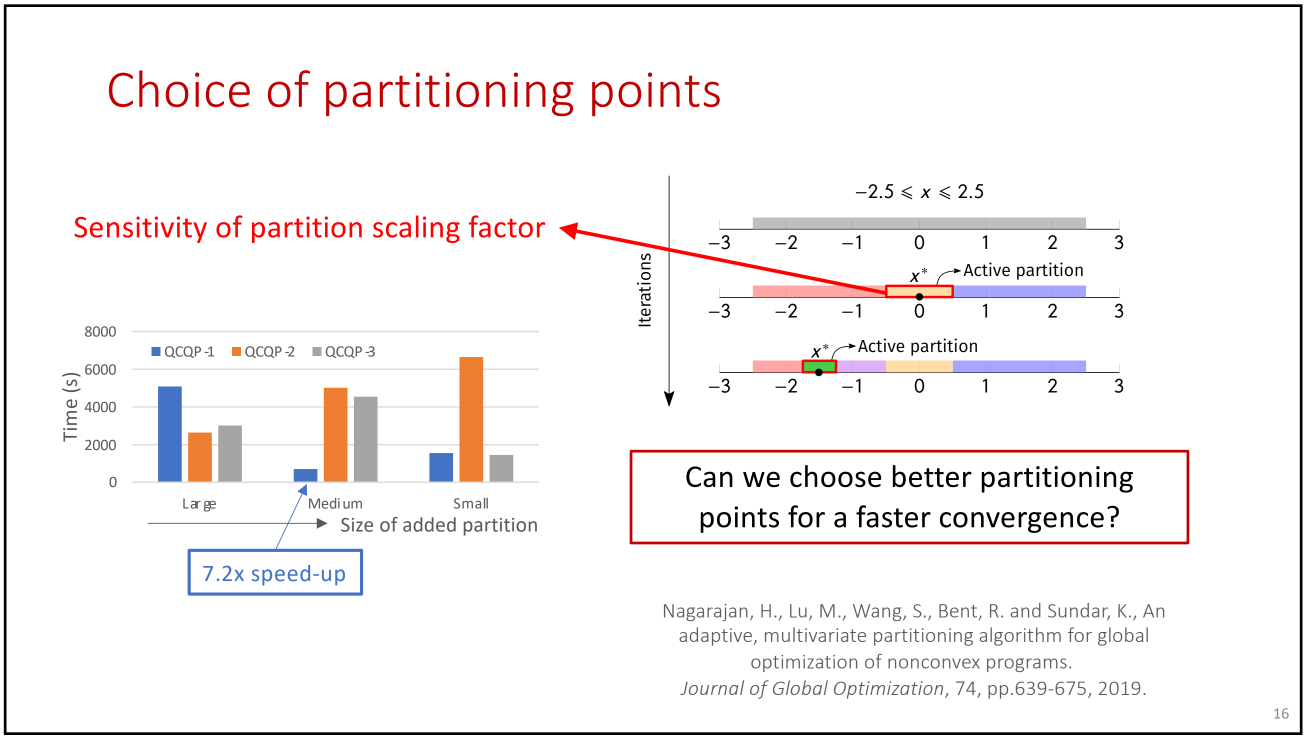
Partitioning-based methods





- Compact search trees
- Complex for implementation!

- Additional partitions result in expensive MIPs.
- Harnesses high-quality MIP solvers.



Choice of partitioning points

Sensitivity of partition scaling factor

Can we choose better partitioning points for a faster convergence?

Nagarajan, H., Lu, M., Wang, S., Bent, R. and Sundar, K., An adaptive, multivariate partitioning algorithm for global optimization of nonconvex programs. *Journal of Global Optimization*, 74, pp.639-675, 2019.

Strong Partitioning and its ML approximation

“Strong Partitioning” (SP) policy

Our Approach:
Choose partitioning policies to maximize the lower bound of the PPR formulation (MIP).

$$p^* \in \arg\max_{p \in P} \ \nu^{PPR}(p) \longleftarrow \text{Set of partitioning points}$$

$$\begin{aligned} \nu^{PPR}(p) := \min \quad & \langle f(\boldsymbol{x}) \rangle^{PPR(p)} \\ s.t. \quad & \langle c_i(\boldsymbol{x}) \rangle^{PPR(p)} = 0, \quad \forall i \in Eq \\ & \langle h_i(\boldsymbol{x}) \rangle^{PPR(p)} \leqslant 0, \quad \forall i \in In \\ & x \in \mathbb{R}^n. \end{aligned}$$

- We currently apply this SP policy only at the 1st iteration of the lower bounding partitioning algorithm (can choose any number of partitioning points).
- Subsequent iterations are still guaranteed to converge to near global optimum.
- Apply local optimization methods to solve SP using generalized gradients.

On solving the SP problem

- The “max-min” SP problem can be formulated as a generalized semi-infinite program.
- Solving it to global optimality may be as hard as solving the original nonconvex QCQP.

We design a local optimization method to solve SP to obtain p^* that yields a tight lower bound, $\nu^{PPR}(p^*)$

For almost every $p \in P$ with $\nu^{PPR}(p) < \nu^*$, the MILP relaxation has a unique partition-solution p^*, and a Clarke generalized gradient of $\nu^{PPR}(p^*)$ can be computed by parametric LP sensitivity.

In the complementary case $\nu^{PPR}(p) = \nu^*$, the lower bound is already tight and no further refinement is needed.

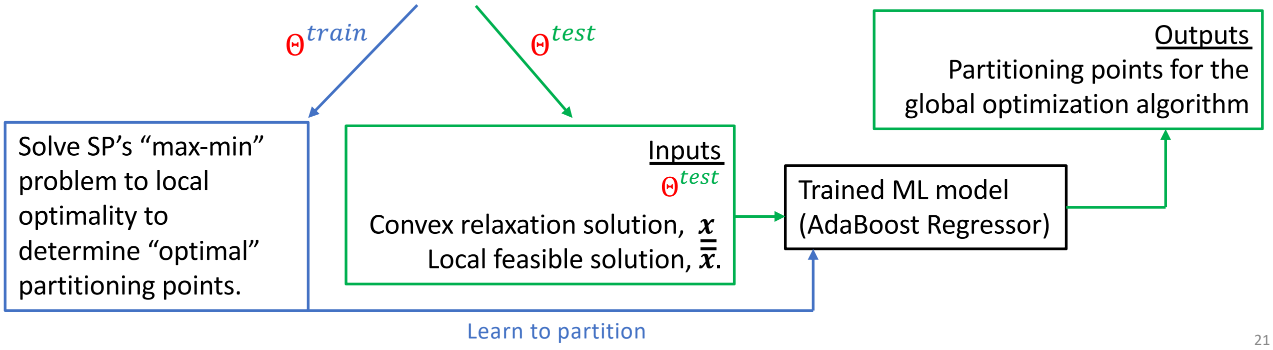
---

1) Mäkelä, M.M., Karmitsa, N. and Wilppu, O., Proximal bundle method for nonsmooth and nonconvex multiobjective optimization. Mathematical modeling and optimization of complex structures, pp.191-204, 2016.

Using learning to imitate the expert policy

Parametric Family (Θ) of nonconvex QCQPs

$$\begin{aligned} \min \quad & f(\boldsymbol{x}, \Theta) \\ s.t. \quad & c_i(\boldsymbol{x}, \Theta) = 0, \quad \forall i \in Eq \\ & h_i(\boldsymbol{x}, \Theta) \leqslant 0, \quad \forall i \in In \\ & x \in \mathbb{R}^n. \end{aligned}$$



Θ^{train}

Θ^{test}

Solve SP's "max-min" problem to local optimality to determine "optimal" partitioning points.

Inputs
Θ^{test}
Convex relaxation solution, $\underline{\boldsymbol{x}}$
Local feasible solution, $\bar{\boldsymbol{x}}$.

Trained ML model (AdaBoost Regressor)

Outputs
Partitioning points for the global optimization algorithm

Learn to partition

Numerical Results

Numerical Experiments

Optimization compute details

- LANL’s Darwin cluster with AMD EPYC 7702 64-core processor, two 2 GHz CPUs, and 60 threads
- MIP relaxation solver: Gurobi v9.1.2
- Nonconvex solver: Ipopt
- Partitioning-based global optimization algorithms:
  - **Alpine.jl:** https://github.com/lanl-ansi/Alpine.jl
  - All pre-solve methods (bound tightening and bound propagation) were *deactivated*
- We apply pre-processing and post-processing steps to reduce training times based on lower bound sensitivities (by eliminating PPs).

Numerical Experiments

Random QCQPs

$$
\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{w}} \quad & c(\Theta)^T\boldsymbol{x} + d(\Theta)^T\boldsymbol{w} \\
s.t. \quad & A(\Theta)\boldsymbol{x} + B(\Theta)\boldsymbol{x} \leqslant b \\
& w_{ij} = x_i x_j, \quad \forall\ (i,j) \in \mathcal{B}, \\
& \boldsymbol{x} \in [0,1]^{d_x}.
\end{aligned}
$$

Test instances

$d_x \in \{10, 20, 50\}$

$5d_x$ bilinear terms

d_x bilinear inequalities

$d_x/5$ linear equalities

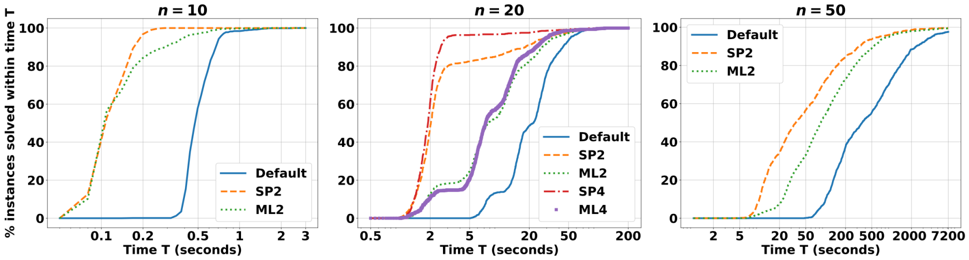
ML model details

- Generate 1000 training samples (varying Θ) with 10-fold cross validation for predictions
- Use Scikit-learn’s AdaBoostRegressor to train Regression Trees

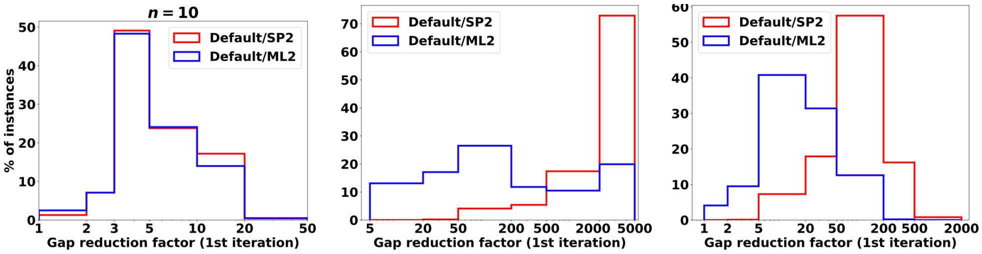
https://github.com/lanl-ansi/Alpine.jl/tree/master/examples/random_QCQPs

QCQP instances

Solution profiles (higher is better)



Histogram of ratios of optimality gaps



Random QCQPs

10-fold average over 1000 instances

Problem Family	Solution Method	Speedup/Slowdown Factor								
		< 0.5	$0.5 - 1$	$1 - 2$	$2 - 5$	$5 - 10$	$10 - 20$	$20 - 50$	> 50	
Bilinear $n = 20$	% Alpine+SP2 inst.	0.2	3.3	7.2	18.2	31.2	29.9	10.0	0.0	49x
	% Alpine+ML2 inst.	3.3	9.8	25.5	39.2	15.3	6.0	0.9	0.0	38x
	% Alpine+SP4 inst.	0.2	0.7	1.3	13.4	32.7	37.1	14.5	0.1	60x
	% Alpine+ML4 inst.	2.8	10.5	23.3	41.4	15.2	5.9	0.9	0.0	29x

2 partitioning points

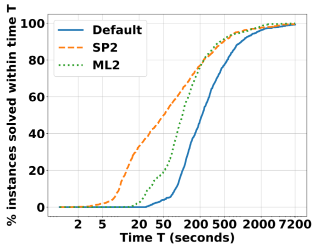
4 partitioning points

↑

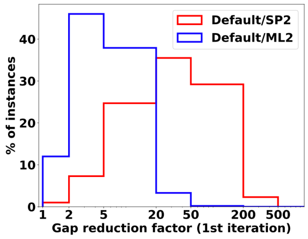
Maximum speedup

Pooling Problem

Solution profiles (higher is better)



Histogram of ratios of optimality gaps



- After the first iteration, Alpine+SP2 closes the effective optimality gap for 45.2% of the instances, whereas default Alpine is unable to close the gap for any of the 1000 instances.
- Run times w.r.t BARON:

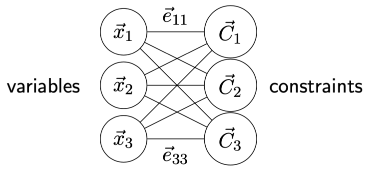
	Average shifted geometric mean (sec.)	Speedup
BARON	441.4	
Alpine (default)	242.8	**1.8x**
Alpine+SP2	66.7	**6.6x**
Alpine+ML2	117.1	**3.8x**

Graph-based
End-to-End
ML framework

Summary of the framework

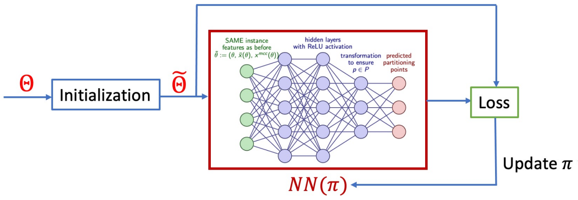
1. Graph-based representation

- Generalizability: Same ML model works for different graph sizes (QCQPs).
- Invariance: Invariant under permutations of variables and constraints.



2. End-to-end (E2E) learning framework

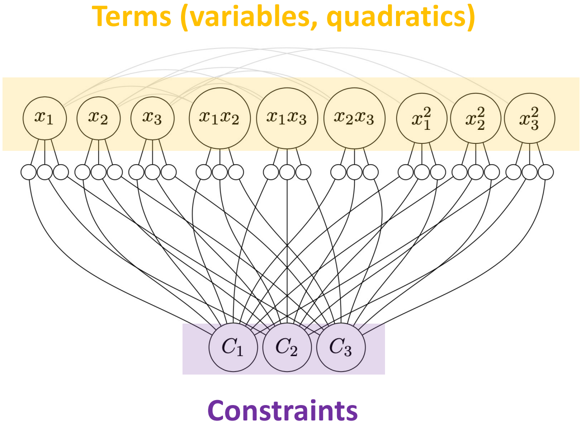
- Decision-focused learning: Integrate downstream decisions into predictive model.
- Tweaked loss function: Novel E2E loss function, with backpropagation challenges.



QCQP as a Computation Graph

Non-convex QCQP

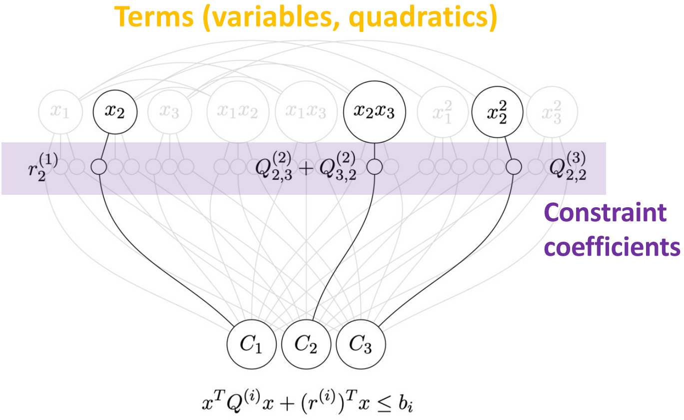
$$\begin{aligned}\min_{x\in[0,1]^n} \quad & x^TQ^{(0)}x + (r^{(0)})^Tx \\ \text{s.t.} \quad & x^TQ^{(1)}x + (r^{(1)})^Tx \le b_1 \\ & x^TQ^{(2)}x + (r^{(2)})^Tx \le b_2 \\ & \vdots \\ & x^TQ^{(N)}x + (r^{(N)})^Tx \le b_N\end{aligned}$$



QCQP as a Computation Graph

Non-convex QCQP

$$\begin{aligned}\min_{x\in[0,1]^n}\quad & x^TQ^{(0)}x+(r^{(0)})^Tx\\ \text{s.t.}\quad & x^TQ^{(1)}x+(r^{(1)})^Tx\le b_1\\ & x^TQ^{(2)}x+(r^{(2)})^Tx\le b_2\\ & \vdots\\ & x^TQ^{(N)}x+(r^{(N)})^Tx\le b_N\end{aligned}$$



QCQP as a Computation Graph

Non-convex QCQP

$$\begin{aligned}\min_{x\in[0,1]^n}\quad & x^TQ^{(0)}x+(r^{(0)})^Tx\\ \text{s.t.}\quad & x^TQ^{(1)}x+(r^{(1)})^Tx\le b_1\\ & x^TQ^{(2)}x+(r^{(2)})^Tx\le b_2\\ & \vdots\\ & x^TQ^{(N)}x+(r^{(N)})^Tx\le b_N\end{aligned}$$

ML model features

Feature	**Nodes**
Coefficient in objective	Variables, quadratics
Cosine similarity with objective	Constraints
Coefficient	Term-constraint edge nodes
Value at presolve	Variables, quadratics, constraints
Value at optimum of initial relaxation	Variables, quadratics, constraints
Relaxation discrepancy ($w_{ij}-x_ix_j$)	quadratics
Is inequality active at presolve?	Constraints
Is inequality active at relaxation?	Constraints

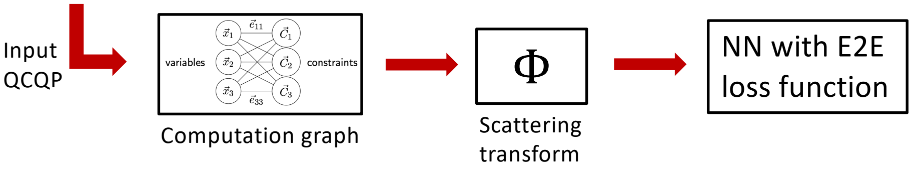
QCQP as a Computation Graph

Non-convex QCQP

$$\begin{aligned} \min_{x\in[0,1]^n} \quad & x^TQ^{(0)}x + (r^{(0)})^Tx \\ \text{s.t.} \quad & x^TQ^{(1)}x + (r^{(1)})^Tx \leq b_1 \\ & x^TQ^{(2)}x + (r^{(2)})^Tx \leq b_2 \\ & \vdots \\ & x^TQ^{(N)}x + (r^{(N)})^Tx \leq b_N \end{aligned}$$

Finally, we apply "Scattering transform" (Φ)

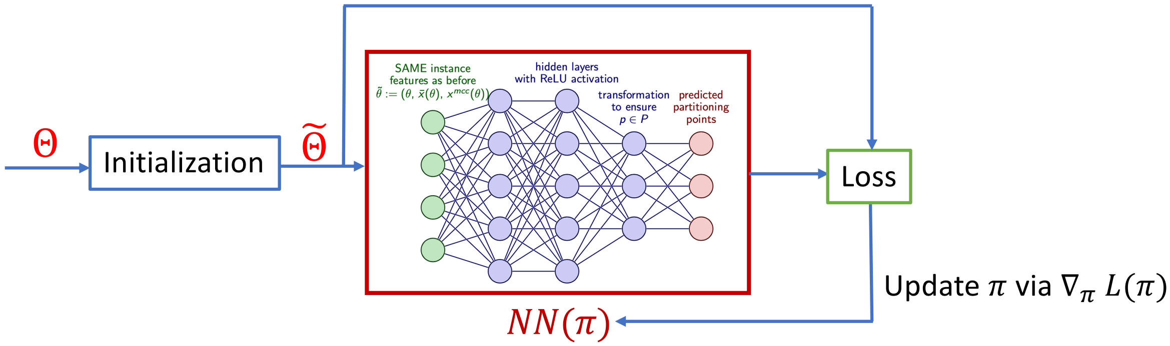
- A non-parametric equivalent of a convolutional NN.
- Construct ReLU transforms and nonlinearities to essentially mix high and low frequency data, capturing convolutional structure.



Mallat, S., 2012. Group invariant scattering. *Communications on Pure and Applied Mathematics*, *65*(10), pp.1331-1398.

End-to-End learning policy

<u>Goal</u> : Directly fit a Neural Network (NN) with weights π for the parametric family.



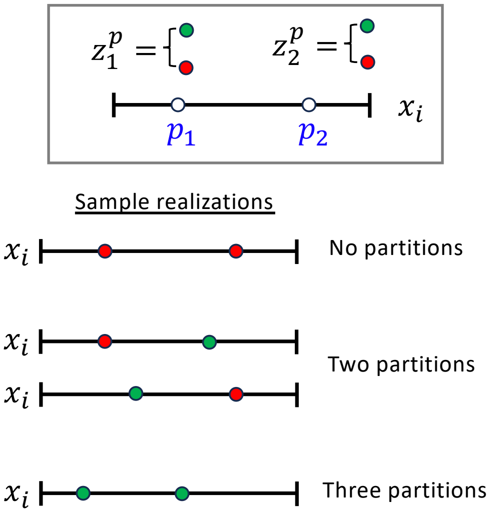
Maximize loss function, promoting sparse partitioning: $L(\pi) = \mathbb{E}_{\Theta}[\nu^{PPR}(p(\pi, \Theta))]$

End-to-End loss function

Lower bound from the MIP relaxation

$$\nu^{PPR}(\Theta, z^p, p)$$

- Θ - Input parameterized QCQP,
- $p \in [0,1]^{2n}$ - Partition points (continuous variables),
- $z^p \in \{0,1\}^{2n}$ - Partition policies (binary variables).



End-to-End loss function

Idealized loss function: $$\mathcal{L}(\Theta, z^p, p) = \underbrace{-\frac{\nu^{PPR}(\Theta, z^p, p)}{|v^*(\Theta)| + \epsilon_{\text{reg}}}}_{\text{Maximize relative lower bound}} + \underbrace{\alpha \|z^p\|_0}_{\text{Promote sparsity}}$$

- $v^*(\Theta)$: Global optimal value/upper-bound of the QCQP.
- $\epsilon_{\text{reg}}, \alpha$: Regularization constants.

Loss function is non-differentiable in both p and z^p.
Relax binary $z^p \in \{0,1\}^{2n}$ to continuous stochastic variables:
$z_i^p \sim \text{Bern}(\tilde{z}_i^p)$ with $\tilde{z}^p \in [0,1]^{2n}$. Take the expectation to define

$$\mathcal{L}_0(\Theta,\ \tilde{z}^p,\ p) \ := \ \mathbb{E}_{z^p \sim \text{Bern}(\tilde{z}^p)}[\,\mathcal{L}(\Theta,\ z^p,\ p)\,]\,,$$

which is smooth in $\tilde{z}^p$ and admits a generalized gradient in p.

Stochastic Partitioning - Gradients

Idealized loss function: $\mathcal{L}(\Theta, z^p, p) = -\dfrac{\nu^{PPR}(\Theta, z^p, p)}{|v^*(\Theta)| + \epsilon_{\text{reg}}} + \alpha \|z^p\|_0$

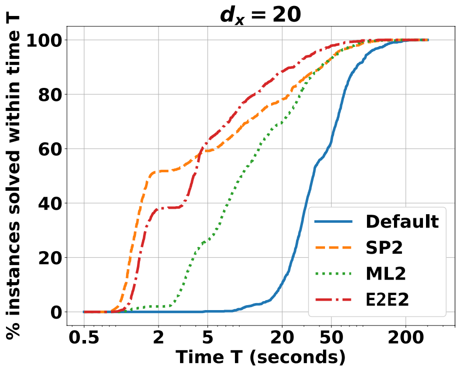
$$\frac{\partial \mathcal{L}_0}{\partial p_i} = -\frac{1}{|v^*(\Theta)| + \epsilon_{\text{reg}}} \cdot \mathbb{E}_{z^p}\left[\frac{\partial \, \nu^{PPR}(\Theta, z^p, p)}{\partial p_i}\right]$$

$$\frac{\partial \mathcal{L}_0}{\partial \tilde{z}_i^p} = \mathbb{E}_{z^p}\left[\frac{\nu^{PPR}(\Theta, z^p, p)}{|v^*(\Theta)| + \epsilon_{\text{reg}}}\middle| z_i^p = 0\right] - \mathbb{E}_{z^p}\left[\frac{\nu^{PPR}(\Theta, z^p, p)}{|v^*(\Theta)| + \epsilon_{\text{reg}}}\middle| z_i^p = 1\right] + \alpha$$

Computing gradients

- For p : Generalized gradient exists and can be computed from any pair of primal and dual solutions of the LP formed by fixing the integers of partitions.
- For $\tilde{z}^p$: Sampling binary realizations from the expectation requires solving a MIP.
- Proposed an efficient epoch-based gradient algorithm to approximate expectations.

Performance of E2E policy



Alpine+SP2: 5.2x
Alpine+ML2: 3.1x
Alpine+E2E2: 6.4x

E2E partitioning policy is very promising!

Generalization performance

Trained on QCQPs with and without quadratic terms with 10 or 20 variables each.
Tested on 50 variable QCQPs.

Problem type	Partition	Shifted GM	Median	Min.	Max.
QCQP $n = 50$	default	319.8	253.3	33.0	6892
QCQP $n = 50$	strong	58.5	52.5	3.12	5375
QCQP $n = 50$	ML	273.3	270.7	18.3	7200
Bilinear $n = 50$	default	326.2	289.0	48.5	7200
Bilinear $n = 50$	strong	43.5	30.9	3.58	7200
Bilinear $n = 50$	ML	189.0	172.3	23.4	5352

Alpine+SP2: 5.5x
Alpine+E2E2: 1.2x

Alpine+SP2: 7.5x
Alpine+E2E2: 1.7x

Reference

Kannan, R., Nagarajan, H. and Deka, D., *"Strong Partitioning and a Machine Learning Approximation for Accelerating the Global Optimization of Nonconvex QCQPs"*. INFORMS Journal on Computing, 2025.

https://doi.org/10.1287/ijoc.2023.0424

Thank you!

harsha@lanl.gov